WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Friday, April 27, 2007

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ		
☐	L13	L12 and(self-modifying code).clm.	2
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
☐	L12	(region near2 memory).clm.	7467
☐	L11	L9 and (region near2 memory)	40
☐	L10	L7 and new partition$	1
☐	L9	L7 and partition$	106
☐	L7	L6 and memory	307
☐	L6	L5 and translat$	308
☐	L5	(self-modif$ code or self modif$ code)	498
☐	L4	L3 and partition$	11
☐	L3	L2 and translat$	32
☐	L2	L1 and (self-modif$ code)	39
☐	L1	717/106-108,136-145,151-157.ccls.	4707

END OF SEARCH HISTORY

Terms used **self modifying code partition overlap** Found **7** of **200,192**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 7 of 7

Relevance scale

1 An experimental implementation of the kernel/domain architecture

Michale J. Spier, Thomas N. Hastings, David N. Cutler
January 1973 **ACM SIGOPS Operating Systems Review , Proceedings of the fourth ACM symposium on Operating system principles SOSP '73**, Volume 7 Issue 4
Publisher: ACM Press

Full text available: pdf(969.98 KB) Additional Information: full citation, abstract, references, citings, index terms

As part of its effort to periodically investigate various new promising concepts and techniques, the Digital Equipment Corporation has sponsored a research project whose purpose it was to effect a limited implementation of a protective operating system framework, based on the kernel/domain architecture which has increasingly been propounded in recent years. The project was carried out in 1972, and its successful completion has led to a substantial number of ...

Keywords: Address space, Domain, Domain incarnation, Kernel, Memory space, Process, Protection

2 Randomized instruction set emulation

Elena Gabriela Barrantes, David H. Ackley, Stephanie Forrest, Darko Stefanović
February 2005 **ACM Transactions on Information and System Security (TISSEC)**, Volume 8 Issue 1
Publisher: ACM Press

Full text available: pdf(374.44 KB) Additional Information: full citation, abstract, references, citings, index terms

Injecting binary code into a running program is a common form of attack. Most defenses employ a "guard the doors" approach, blocking known mechanisms of code injection. *Randomized instruction set emulation* (RISE) is a complementary method of defense, one that performs a hidden randomization of an application's machine code. If foreign binary code is injected into a program running under RISE, it will not be executable because it will not know the proper randomization. The pape ...

Keywords: Automated diversity, randomized instruction sets, software diversity

3 Thread-level speculation: Speculative thread decomposition through empirical optimization

Troy A. Johnson, Rudolf Eigenmann, T. N. Vijaykumar
March 2007 **Proceedings of the 12th ACM SIGPLAN symposium on Principles and practice of parallel programming PPoPP '07**
Publisher: ACM Press

Full text available: pdf(511.96 KB) Additional Information: full citation, abstract, references, index terms

Chip multiprocessors (CMPs), or multi-core processors, have become a common way of reducing chip complexity and power consumption while maintaining high performance. *Speculative* CMPs use hardware to enforce dependence, allowing a parallelizing compiler to generate multithreaded code without needing to prove independence. In these systems, a sequential program is decomposed into threads to be executed in parallel; dependent threads cause performance degradation, but do not affect correc ...

Keywords: chip multiprocessor, decomposition, empirical search, multi-core, thread-level speculation

4 Cache Memories

Alan Jay Smith
September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3
Publisher: ACM Press
Full text available: pdf(4.61 MB) Additional Information: full citation, references, citings, index terms

5 Articles: Division of Labor in Embedded Systems

Ivan Goddard
April 2003 **Queue**, Volume 1 Issue 2
Publisher: ACM Press
Full text available: html(37.05 KB) Additional Information: full citation, index terms

6 Selection conditions in main memory

Kenneth A. Ross
March 2004 **ACM Transactions on Database Systems (TODS)**, Volume 29 Issue 1
Publisher: ACM Press
Full text available: pdf(296.54 KB) Additional Information: full citation, abstract, references, citings, index terms

We consider the fundamental operation of applying a compound filtering condition to a set of records. With large main memories available cheaply, systems may choose to keep the data entirely in main memory, in order to improve query and/or update performance.The design of a data-intensive algorithm in main memory needs to take into account the architectural characteristics of modern processors, just as a disk-based method needs to consider the physical characteristics of disk devices. An importa ...

Keywords: Branch misprediction

7 Run-time generation of HPS microinstructions from a VAX instruction stream

Y. N. Patt, S. W. Melvin, W. M. Hwu, M. C. Shebanow, C. Chen
December 1986 **ACM SIGMICRO Newsletter , Proceedings of the 19th annual workshop on Microprogramming MICRO 19**, Volume 17 Issue 4
Publisher: ACM Press
Full text available: pdf(808.93 KB) Additional Information: full citation, abstract, references, citings, index terms

The VAX architecture is a popular ISP architecture that has been implemented in several different technologies targeted to a wide range of performance specifications. However, it has been argued that the VAX has specific characteristics which preclude a very high performance implementation. We have developed a microarchitecture (HPS) which is specifically intended for implementing very high performance computing engines. Our model of execution is a restriction on fine granularity data flow. ...

Web Results **1** - **10** of about **389** for **"self-modifying code" partition overlap translation memory**. (**0.31** seconds)

[PPT] A Reflective Middleware Framework for Communication in Dynamic ...
File Format: Microsoft Powerpoint - View as HTML
User processes then held in high **memory**. Single **partition** allocation. Relocation register scheme used ... Reentrant (non **self-modifying) code** can be shared. ...
www.ics.uci.edu/~ics143/lectures/oslecture10-13.ppt - Similar pages

Windows 95b MBR
Discussion of the Windows 95 (OSR2) MBR, new **partition** types and the use of ... copy of boot record at Sect 06) **Self-modifying code** if BIOS supports Int 13 ...
home.att.net/~rayknights/pc_boot/w95b_mbr.htm - 52k - Cached - Similar pages

[PDF] AN EXPERIMENTAL IMPLEMENTATION OF THE KERNEL/DOMAIN ARCHITECTURE ...
File Format: PDF/Adobe Acrobat
sive **memory** space **partition**, then these computations are said to be ... Segment which is an execute-only body of pure, non-**self modifying code**, whose single ...
portal.acm.org/citation.cfm?doid=957195.808043 - Similar pages

Postsilicon Validation Methodology for Microprocessors
It is useful to test the algorithm with different **memory** types for data and code ... It is important to keep an appropriate coverage **overlap** with randomized ...
doi.ieeecomputersociety.org/10.1109/54.895008 - Similar pages

[PDF] Postsilicon validation methodology for microprocessors - Design ...
File Format: PDF/Adobe Acrobat
Do S2 length loads into L2 from **memory** address M1+2S3-S2 ... serializing events, **self-modifying code** related events etc.} used. when Events enable is 1 ...
ieeexplore.ieee.org/iel5/54/19366/00895008.pdf?arnumber=895008 - Similar pages

[PDF] **Memory** Management **Memory** Management
File Format: PDF/Adobe Acrobat - View as HTML
overlapping. —. Some dedicated mechanisms are supported by special addressing ... associative **memory** or **translation** look-aside buffers ...
www.it.uc3m.es/tao/**Memory**Management_EN.pdf - Similar pages

Improving Application Performance Through System Call Composition
Also, if we use two non-**overlapping** segments for function code and function data, concerns due to **self modifying code** vanish automatically. ...
www.fsl.cs.sunysb.edu/docs/cosy-perf/ - 86k - Cached - Similar pages

Latest Patents
10, 20070079102, Assigning a processor to a logical **partition** ... 3, 7185337, Efficient locking for thread-safe **self-modifying code** ...
www.latestpatents.com/category/ibm/feed/ - 388k - Cached - Similar pages

[PDF] A Survey on Virtualization Technologies
File Format: PDF/Adobe Acrobat - View as HTML
translation unit. **Self-modifying code** is a special challenge in x86 ... the OS produces a **partition** per virtual machine on demand that is a replica of the ...
www.ecsl.cs.sunysb.edu/tr/TR179.pdf - Similar pages

[PDF] **Memory** Systems
File Format: PDF/Adobe Acrobat - View as HTML
Key Idea: A **memory** location may reside in multiple places ... **self-modifying code** can cause problems ... avoding **translation** during L1 indexing (later) ...

Search Results 1 - 10 of about 41 for **"self-modifying code" partition overlap translation memory** - 0.92 sec. (About this page

vw.rh.edu/~steves/opsys/cStore9mem.ppt (MICROSOFT POWERPOINT)
oy the **memory** unit, after **translation** from logical space ... Text (read only, reentrant, non
lf **modifying) code** for editor is the same for all users. ...
vw.rh.edu/~steves/opsys/cStore9mem.ppt - 885k - View as html - More from this site

ındomized Instruction Set Emulation (PDF)
code sequences loaded into emulator **memory** from the local ... **self-modifying code** as an
desirable programming practice and agree with Valgrind's ...
vw.cs.unm.edu/~forrest/publications/rise-tissec.pdf - 301k - View as html - More from this site

Reflective Middleware Framework for Communication in Dynamic
ıvironments (PDF)
ıltiple **partition** Allocation. Hole - block of available **memory**; holes of various sizes ...
entrant (non **self-modifying) code** can be shared. ...
vw.ics.uci.edu/~ics143/lectures/oslecture10-13.pdf - 446k - View as html - More from this site

Survey on Virtualization Technologies (PDF)
ıses 16MB system **memory** for use as a "**translation** ... **translation** unit. **Self-modifying**
de is a special challenge in x86 emulation because no ...
vw.ecsl.cs.sunysb.edu/tr/TR179.pdf - 594k - View as html - More from this site

ılection Conditions in Main **Memory** (PDF)
partition a compound filter into nested conjunctions and disjunctions. The ... Run-time
de specialization is different from **self-modifying code**. With self ...
vw.cs.columbia.edu/~kar/pubsk/selcondsTODS.pdf - 297k - View as html - More from this site

emory Systems (PDF)
lf-modifying code can cause problems. caches should be split if simultaneous I and D
cesses ... **Partition** cache frames into. • equivalence classes ...
vw.eecg.toronto.edu/~moshovos/ACA06/lecturenotes/**memory**.pdf - 401k - View as html -
re from this site

indows 95b MBR
scussion of the Windows 95 (OSR2) MBR, new **partition** types and the use of Interrupt 13
ensions ... at Sect 06) **Self-modifying code** if BIOS supports Int ...
ne.att.net/~rayknights/pc_boot/w95b_mbr.htm - 52k - Cached - More from this site

el® IA-64 Architecture Software Developer's Manual (PDF)
aning that the size of the clean register stack **partition** is always zero. ... data and
truction caches, virtual **memory translation** structures, and more. ...
vw.cs.umbc.edu/help/architecture/24532001.pdf - 829k - View as html - More from this site

ındomized Instruction Set Emulation (PDF)
consider arbitrary **self-modifying code** as an undesirable programming ... the curves
erlap. A second observation (not shown) is that **memory** density has ...
vw.cs.unm.edu/~darko/papers/p3-barrantes.pdf - 374k - View as html - More from this site

el® Itanium' processor reference manual for software development (PDF)
he size of the clean register stack **partition** is always zero. ... processor data and instruction
ches, virtual **memory translation** structures, and more. ...
vresource.hp.com/drc/STK/docs/refs/24532003s.pdf - 1567k - View as html - More from this